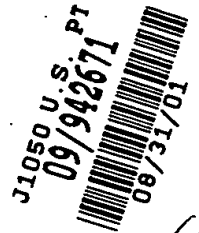


日 本 国 特 許 庁  
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日  
Date of Application:

2000年 9月 1日

出 願 番 号  
Application Number:

特願2000-265785

出 願 人  
Applicant(s):

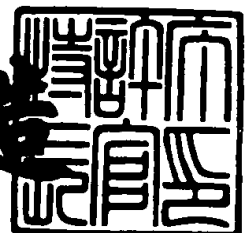
日本電気株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 5月31日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3050241

#3

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant: Toshihiko OKAMURA  
Title: ERROR-CORRECTING CODE  
DECODING METHOD AND  
ERROR-CORRECTING CODE  
DECODING APPARATUS



Appl. No.: Unassigned  
Filing Date: August 31, 2001  
Examiner: Unassigned  
Art Unit: Unassigned

**CLAIM FOR CONVENTION PRIORITY**

Commissioner for Patents  
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application:

- JAPAN Patent Application No. 2000-265785 filed 09/01/2000.

Respectfully submitted,

Date: August 31, 2001

FOLEY & LARDNER  
Washington Harbour  
3000 K Street, N.W., Suite 500  
Washington, D.C. 20007-5109  
Telephone: (202) 672-5407  
Facsimile: (202) 672-5399

By Thomas S. Blumenthal Reg. No. 43438

*for* David A. Blumenthal  
Attorney for Applicant  
Registration No. 26,257

【書類名】 特許願

【整理番号】 35001020

【提出日】 平成12年 9月 1日

【あて先】 特許庁長官 殿

【国際特許分類】 H03M 7/00

【発明者】

    【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

    【氏名】 岡村 利彦

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100088328

    【弁理士】

    【氏名又は名称】 金田 暢之

    【電話番号】 03-3585-1882

【選任した代理人】

    【識別番号】 100106297

    【弁理士】

    【氏名又は名称】 伊藤 克博

【選任した代理人】

    【識別番号】 100106138

    【弁理士】

    【氏名又は名称】 石橋 政幸

【手数料の表示】

    【予納台帳番号】 089681

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9710078

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 誤り訂正符号復号方法及び装置

【特許請求の範囲】

【請求項 1】 所定の復号処理を繰り返すことで受信語を復号する誤り訂正符号復号方法であって、

前記受信語をブロックに分割し、

前記ブロックへの分割方法は、前記所定の復号処理の繰り返しが何回目かであるかに応じて決定し、

前記ブロック単位で、前記受信語及び前記受信語の情報シンボルの信頼度に相当する事前分布値から前記情報シンボルの軟出力値をそれぞれ生成する誤り訂正符号復号方法。

【請求項 2】 前記ブロックの基準の大きさである時点数を  $B$ 、ブロックの種類数を  $n$ 、現時点の復号処理における繰り返し回数を  $k$  回目としたとき、

前記ブロックへの分割法は、前記ブロックの中で最初のブロックの大きさを、

$B(n+1-k)/n$ 、

とし、最終ブロックを除く残りのブロックの大きさを  $B$  とする請求項 1 記載の誤り訂正符号復号方法。

【請求項 3】 現時点の復号処理における繰り返し回数が奇数回目または偶数回目のいずれか一方で、

最初のブロックの大きさを所定の大きさの  $1/2$  に設定する請求項 1 記載の誤り訂正符号復号方法。

【請求項 4】 所定の復号処理を繰り返すことで受信語を復号する誤り訂正符号復号装置であって、

前記受信語を格納する受信値格納装置と、

前記受信語の情報シンボルの信頼度に相当する事前分布値をそれぞれ保持する事前分布値格納装置と、

前記受信格納装置及び前記事前分布値格納装置から前記ブロック単位でデータを読み出す入力制御装置と、

前記入力制御装置で読み出された前記受信語と前記事前分布値から前記情報シ

ンボルの軟出力値を前記ブロック単位で生成する軟出力生成装置と、

前記入力制御装置に、最初のブロックの大きさを現時点の復号処理における繰り返し数に応じてそれぞれ異なった値に設定させる制御情報生成装置と、  
を有する誤り訂正符号復号装置。

【請求項5】 前記制御情報生成装置は、

前記ブロックの基準の大きさである時点数を  $B$ 、ブロックの種類数を  $n$ 、現時点の復号処理における繰り返し回数を  $k$  回目としたとき、

前記ブロックへの分割法は、前記ブロックの中で最初のブロックの大きさを、  
 $B(n+1-k)/n$ 、

とし、最終ブロックを除く残りのブロックの大きさを  $B$  とする請求項4記載の誤り訂正符号復号装置。

【請求項6】 前記制御情報生成装置は、

前記入力制御装置に、現時点の復号処理における繰り返し数が奇数回目または偶数回目のいずれか一方で、最初のブロックの大きさを所定の大きさの  $1/2$  に設定させる請求項4記載の誤り訂正符号復号装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は誤り訂正符号により符号化されたデータを復号する誤り訂正符号復号装置に関し、特にターボ符号に代表される接続符号の復号処理に用いて好適な誤り訂正符号復号方法及び装置に関する。

【0002】

【従来の技術】

誤り訂正符号化技術は、データ伝送時に通信路上でビット反転等のエラーが発生した場合でもデータの符号化・復号操作によって誤りのないデータを高い確率で再生する技術であり、無線通信やデジタル記録媒体を始めとして現在広く利用されている。符号化は送信するデータ（情報系列）に冗長性を付加した符号語へ変換する操作であり、復号はその冗長性を利用してエラーが混入した符号語（受信語）から元のデータを推定する操作である。

## 【0003】

通常、復号装置には前述のように符号語、またはそれを生成するための情報系列が与えられる。これらの受信語から重み付きで各情報ビットを推定する復号方法が知られており、このような復号方法として軟出力復号法がある。

## 【0004】

最適な軟出力復号法は、各情報シンボルが符号語を構成するシンボル列であるという制約条件の下、受信語に対する条件付き確率を出力する復号方法であり、事後確率復号法と呼ばれる。符号化対象の情報シンボルが2値(0, 1)の場合、事後確率復号では下記式(1)で示される $L(u(k))$ を生成すればよい。

## 【0005】

【数1】

$$L(u(k)) = \log(P(u(k)=0|Y) / P(u(k)=1|Y)) \quad \dots(1)$$

## 【0006】

ここで、 $u(k)$ は $k$ 番目の情報ビット、 $Y$ は受信語、 $P(u(k)=b|Y)$  ( $b=0, 1$ )は受信語 $Y$ の下で $u(k)=b$ となる条件付き確率である。

## 【0007】

特に、状態数の少ないトレリス線図で表すことが可能な符号に対する復号方法として、比較的計算量の少ない事後確率復号方法が知られている。事後確率復号方法を実現するアルゴリズムは通常の畳込み符号の最尤復号法として知られているヴィタビ(Viterbi)復号の処理を複雑にした手法であり、BCJRアルゴリズムまたはMAPアルゴリズムと呼ばれている。MAPアルゴリズムについては、例えば“Optimal decoding of linear codes for minimizing symbol error rate”(IEEE Transaction on Information Theory, pp.284-287, 1974)に詳述されている。

## 【0008】

MAPアルゴリズムは、受信語 $Y$ を直接“0”または“1”に判定した硬判定結果ではなく、各ビットの判定に信頼度を付加した値、つまり軟判定結果(軟入力値)を出力する復号方法であり、まとめて軟入力軟出力復号法と呼ばれている

。以下、情報シンボルを2値として説明するが多値の場合にも容易に拡張可能である。

#### 【0009】

事後確率復号法では、式(1)の $L(u(k))$ そのものではなく、その近似値を求めることでMAPアルゴリズムの計算量を削減する種々の方法が検討されている。Max-Log-MAPアルゴリズムやSOVA(soft-output Viterbi algorithm)等がその代表的な手法であり、これらのアルゴリズムについては、例えば“A Comparison of optimal and sub-optimal MAP decoding algorithms”(ICC' 95, pp.1009-1013, 1995)で詳述されている。

#### 【0010】

ところで、1993年、C. Berrouらによってターボ符号と呼ばれる高性能な符号が提案された。ターボ符号に関しては、例えば“Near Shannon limit error-correcting coding and decoding: Turbo codes”(ICC' 93, pp.1064-1070, 1993)で詳述されている。

#### 【0011】

図7はターボ符号器、復号器の代表的な構成を示す図であり、同図(a)はターボ符号器のブロック図、同図(b)はターボ復号器のブロック図である。図7(a)に示すように、ターボ符号器は、不図示の遅延素子及び排他的論理和を含むフィードバックループが形成された組織畳込み符号化器がインターリーバを介して2個並列に接続された構成である。

#### 【0012】

第1の組織畳込み符号化器100及び第2の組織畳込み符号化器101による畳込み符号化処理では、通常メモリ数4以下の符号が使用される。インターリーバ102は、符号化対象である情報系列の各ビットを所定の規則で並び替える装置であり、インターリーバ102の処理方式により符号化性能が大きく左右される。なお、ターボ符号器からは情報系列と共に冗長系列であるパリティ1及びパリティ2が出力される。

#### 【0013】

一方、図7(b)に示すように、ターボ復号器は、図7(a)に示したターボ



符号器の構成に対応して2つの復号器を有する構成である。第1の復号器103は第1の組織畳込み符号化器100に対応した復号器であり、第2の復号器105は第2の組織畳込み符号化器101に対応した復号器である。

#### 【0014】

ターボ復号の特徴は第1の復号器103及び第2の復号器105で上述した軟入力軟出力復号法を用いることである。第1の復号器103は、第2の復号器105で生成された各情報ビット系列に対する軟出力により受信データに重みを付与して復号を行う。

#### 【0015】

第1の復号器103に入力される、第2の復号器105で生成した軟出力は、式(1)の値 $L(u(k))$ そのものではなく、 $L(u(k))$ から算出される下記式(2)で表される外部情報と呼ばれる値 $Le(u(k))$ である。

#### 【0016】

【数2】

$$Le(u(k)) = L(u(k)) - C \cdot y(k) - La(u(k)) \quad \dots(2)$$

#### 【0017】

ここで、 $y(k)$ は情報ビット $u(k)$ に対する受信値、 $La(u(k))$ は第2の復号器105から与えられる値、 $C$ は通信路のSN比で決まる定数である。なお、実際の $Le(u(k))$ は、第1の復号器103における情報ビット系列の順序に一致するようにデインターリーバ106により並び替えられる。

#### 【0018】

一方、第1の復号器103で式(2)を用いて算出された $Le(u(k))$ は、第2の復号器105で事前分布値 $La(u(k))$ として使用される。このときも実際の $Le(u(k))$ は、第2の復号器105における情報ビット系列の順序に一致するようにインターリーバ104により並び替えられる。

#### 【0019】

このように二つの復号器の間で外部情報を順次更新する繰り返し復号処理を行うことがターボ復号の特徴である。この繰り返し処理の度に復号化誤り率が順次

改善され、通常10回程度の繰り返し処理で十分となる。繰り返し復号処理の結果、最終的に得られた軟出力値 $L(u(k))$ を硬判定することにより復号データが得られる。

#### 【0020】

次に、MAPアルゴリズム（BCJRアルゴリズム）について更に詳しく説明する。

#### 【0021】

MAPアルゴリズムは、最尤復号法としてよく知られたヴィタビアルゴリズムと同様にトレリス符号を利用したアルゴリズムである。トレリス符号は、符号系列をトレリス線図で表現することが可能であり、各時点にそれぞれ複数の状態が対応する。隣接する時点の状態は符号構造から定まるパスでそれぞれ連結され、パスには符号語のビット列が対応する。特に（時不変の）畳込み符号は時点に依らないトレリス構造を有する特徴があり、各時点における状態数が大きくなければトレリス符号を利用したアルゴリズムを実行することが容易である。

#### 【0022】

MAPアルゴリズムは大きく分けて次の3種類の処理から構成される：

（a）フォワード処理：

トレリス符号の先頭から各時点の各状態へ到達する確率を算出する。

（b）バックワード処理：

トレリス符号の終端から各時点の各状態へ到達する確率を算出する。

（c）軟出力生成処理：

（a）、（b）の処理結果を用いて各時点における情報ビットの事後確率比を算出する。

#### 【0023】

畳込み符号の状態集合を $S = [0, 1, \dots, |S| - 1]$ 、時点 $t$ 、状態 $s$ におけるフォワード処理、バックワード処理で算出される値をそれぞれ $\alpha(t, s)$ 、 $\beta(t, s)$ と表す。また、状態 $s$ から状態 $s'$ への時点 $t$ における遷移の確率を $\gamma(t, s, s')$ と表す。 $\gamma(t, s, s')$ は、状態 $s$ から状態 $s'$ への遷移に対応する符号語と受信値との間の尤度として計算することが可能であり、白色ガウス

通信路では  $\gamma(t, s, s')$  を通信路の SN 比を用いて容易に計算することができる。

【0024】

このときフォワード処理およびバックワード処理は次のように 1 時点前あるいは 1 時点後の値を用いて実行される：

(a) フォワード処理：

【0025】

【数 3】

$$\alpha(t, s) = \sum \alpha(t-1, s') \gamma(t, s', s).$$

【0026】

(b) バックワード処理：

【0027】

【数 4】

$$\beta(t, s) = \sum \beta(t+1, s') \gamma(t+1, s, s').$$

【0028】

ここで、これらの式の和 ( $\Sigma$ ) はすべての状態  $s'$  についてとることを意味する。また、時点  $t$  の事後確率比は、 $\alpha(t-1, s)$ 、 $\gamma(t, s, s')$ 、 $\beta(t, s)$  を用いて算出することができる。

【0029】

Max-log-MAP アルゴリズムは、上記 (a)、(b) の処理において、 $\alpha$ 、 $\beta$ 、 $\gamma$  の対数値  $\alpha'$ 、 $\beta'$ 、 $\gamma'$  をとって演算処理を積から和に変更し、さらに  $\Sigma$  を最大値を求める演算に変更する。つまり Max-log-MAP アルゴリズムでは (a)、(b) が下記 (a')、(b') に変更される。

(a') フォワード処理：

【0030】

【数 5】

$$\alpha'(t, s) = \max\{\alpha'(t-1, s') + \gamma'(t, s', s)\}.$$

【0 0 3 1】

(b' ) バックワード処理:

【0 0 3 2】

【数 6】

$$\beta'(t, s) = \max\{\beta'(t+1, s') + \gamma'(t+1, s, s')\}.$$

【0 0 3 3】

ここで、最大値を求める処理  $\max$  はすべての状態  $s'$  についてとることを意味する。Max-log-MAP アルゴリズムでは MAP アルゴリズムで必要であった積の演算処理が不要になることが大きな特徴である。

【0 0 3 4】

(a' ) の処理は通常のヴィタビアルゴリズムにおける ACS (Add-Compare-Select) 処理に相当し、(b' ) の処理は ACS 処理をトレリス符号の終結部からトレリス符号の始点へ向かって行う逆向きの処理に相当する。

【0 0 3 5】

このように、MAP アルゴリズムや Max-log-MAP アルゴリズムでは、トレリス符号の各時点、各状態における  $\alpha$ 、 $\beta$  の値を保存しておく必要がある。

【0 0 3 6】

また、ターボ符号では状態数は小さいが大きな符号長(情報ビット数 500 程度以上)を扱うため、実装上、トレリス符号の任意の一部を利用して順次復号できるようにする必要がある。これはパスメモリと呼ばれる一定時点数以下のトレリス符号を利用して実行可能であり、ヴィタビアルゴリズムでも適用されている手法である。

【0 0 3 7】

図 8 は従来の誤り訂正符号復号装置の構成を示すブロック図である。

## 【 0 0 3 8 】

図 8 において、事前分布格納装置 7 1 は、受信値と関係なく与えられる符号語のビットまたはシンボルの信頼度情報を格納する装置であり、ターボ復号における信頼度情報は、他方の復号器で生成される各情報ビットの外部情報  $L_e(u(k))$  に相当する。入力制御装置 7 2 は、受信値格納装置 7 0 と事前分布格納装置 7 1 から必要な時点数分のデータを読み出し、軟出力生成装置 7 3 に供給する。軟出力生成装置 7 3 は、MAP アルゴリズムや  $\text{Max-log-MAP}$  アルゴリズム等の軟入力軟出力復号法に基づいて軟出力を生成する。この際、これらアルゴリズムではバックワード処理が必要になる点が従来のヴィタビアルゴリズムと大きく異なっている。

## 【 0 0 3 9 】

初期時点から順次復号を行う自然な状況を想定した場合、フォワード処理はトレリスの始点から開始され、順次更新されていくため問題はない。しかしながら、バックワード処理はトレリス終結点から処理を開始するため、トレリスの途中においてはバックワード処理をどこから始めるか、またその初期値をどのように設定するかという問題が発生する。

## 【 0 0 4 0 】

この問題を解決するためにスライディングウィンドウ法と呼ばれる方法が提案されており、例えば “Soft-output decoding algorithms for continuous decoding of parallel concatenated convolutional codes” (ICC' 96, pp.112-117, 1996) に詳述されている。スライディングウィンドウ法は、時点  $t$  の軟出力を算出するときに、時点  $t + T$  ( $T$  は  $t$  に対して十分大きな時点数 ( $T > 0$ )) の各状態を等確率としてバックワード処理を開始する方法である。

## 【 0 0 4 1 】

スライディングウィンドウ法を利用し、パイプライン化により逐次軟出力が得られるハードウェアアーキテクチャは、例えば “VLSI architectures for turbo codes” (IEEE Transactions on VLSI systems, pp.369-379, 1999) で提案されている。

## 【 0 0 4 2 】

このアーキテクチャでは、入力制御装置 7 2 により受信値格納装置 7 0 及び事前分布格納装置 7 1 から 1 時点分ずつデータを読み込み、パスメモリは T 時点分で 1 時点毎にスライドしていく。したがって、1 情報シンボルあたり T 時点数分のバックワード処理を行うことになる。通常、バックワード処理はトレリスの終結点から初期時点まで 1 回だけ行えばよいので 1 情報シンボルあたり 1 時点数分のバックワード処理を行えば十分であることに注意する。

#### 【 0 0 4 3 】

また、T 時点ずらしたバックワード処理を並列に動作させることにより逐次軟出力を得る方法が “An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes” (IEEE Journal on Selected Areas, pp.260-264,1998) に記載されている。この方法では 1 情報シンボル当りのバックワード処理が 2 時点数分となる。このようにトレリス符号の一部の時点を利用して軟出力復号を行う場合、バックワード処理が増加する。

#### 【 0 0 4 4 】

以上説明した復号方法はトレリス符号の一部のみを用いることによって生じたバックワード処理の増加を並列度の高いアーキテクチャを用いて解決する方法である。一方、DSP のように並列度の低いアーキテクチャでトレリス符号の一部を用いて復号する場合には、1 情報シンボルあたりのバックワード処理の計算量を少なくすることが復号処理速度の改善に必要である。

#### 【 0 0 4 5 】

この場合、符号語を複数のブロックに分割し、ブロック毎に軟出力を生成する処理が有効である。なお、以下、軟出力を生成する時点数の単位 (= ブロックの大きさの基準値) を B と定義する。

#### 【 0 0 4 6 】

図 9 はこの制御の流れを示す図であり、ブロック毎に軟出力を生成する従来の誤り訂正符号復号装置の処理手順を示すフローチャートである。

#### 【 0 0 4 7 】

図 9 に示すように、従来の誤り訂正符号復号器では、まずトレリス終結部を除いた残りの受信データの時点数 (残データ時点数) R を確認する (ステップ D 1)

。そして、残データ時点数  $R$  が  $B$  以上であった場合はブロックの大きさを  $B$  に設定し(ステップ D 2)、残データ時点数  $R$  が  $B$  よりも小さかった場合はブロックの大きさを  $R$  に設定する(ステップ D 3)。したがって、ステップ D 3 ではトレリス終結部から通常のバックワード処理を行うことになる。

## 【 0 0 4 8 】

図 1 0 は図 9 に示したフローチャートにしたがって受信データが復号処理される様子を示す模式図である。なお、ブロックには、初期時点から終結時点に向かって順にブロック 1, 2, ... と番号を付与するものとする。

## 【 0 0 4 9 】

図 1 0 に示すように、従来の誤り訂正符号復号器では、まず最初にブロック 1 に対してフォワード処理  $F(1)$  及びバックワード処理  $B(1)$  がそれぞれ実行される。フォワード処理の初期値はトレリス符号の始点に対応するので問題ないが、バックワード処理の初期値は前述したように各状態が等確率とする。このときブロック境界における  $\beta$  が有意な値をとるためには、 $T$  時点数分のマージンをとった  $(B + T)$  時点からバックワード処理を開始する。つまり、 $B$  時点における受信データを復号するためには、入力制御装置 7 2 は、ブロックの境界で  $(B + T)$  時点分の受信データを受信値格納装置 7 0 及び事前分布格納装置 7 1 から軟出力生成装置 7 3 に供給する必要がある。そして、ブロック 1 に対するフォワード処理  $F(1)$  及びバックワード処理  $B(1)$  が終了したら軟出力生成処理  $G(1)$  を実行する。

## 【 0 0 5 0 】

ブロック 1 に対する全ての処理が終了したら、ブロック 2 に対する処理に移行する( $F(2)$ 、 $B(2)$ 、 $G(2)$ )。このときフォワード処理  $F(2)$  の初期値はブロック 1 の終了時点  $B$  における  $\alpha$  の値をそのまま使用し、バックワード処理  $B(2)$  の初期値は  $(2B + T)$  時点から各状態が等確率であるとして開始される。この方法では 1 情報シンボルあたりのバックワード処理が  $(1 + T/B)$  時点数分となる。また、フォワード処理と軟出力生成処理に関しては 1 情報シンボルあたり 1 時点数分となる。

## 【 0 0 5 1 】

## 【発明が解決しようとする課題】

図 9 及び図 1 0 に示した従来の復号方法では、1 情報シンボルあたりのバックワード処理が  $(1 + T/B)$  時点数分となり、バックワード処理のマージン時点数  $T$  をブロックの大きさ  $B$  に対して小さく設定することでバックワード処理の計算量を削減できる。

## 【0 0 5 2】

装置規模の増大を防ぐためにはブロックの大きさ  $B$ 、マージン時点数  $T$  の値を小さくすることが望ましい。しかしながら、バックワード処理のマージン時点数  $T$  の値を小さくすることは復号性能の劣化につながる。例えば、メモリ数 3 の畳込み符号を接続したターボ符号用の復号器に軟入力軟出力復号を適用した場合、 $T = 24 \sim 30$  で復号性能の劣化が認められなくなるが、 $T = 15$  では復号フレーム誤り率 0.0001 で 0.4 dB 以上の符号化利得の劣化が生じることが実験により確かめられている。

## 【0 0 5 3】

フォワード処理、バックワード処理及び軟出力生成処理をそれぞれ同一の計算量とみなすと、フォワード処理、軟出力生成処理は 1 情報シンボルあたり 1 時点数分、バックワード処理は 1 情報シンボルあたり  $(1 + T/B)$  時点数分の処理のため、 $T$ 、 $B$  の値はトータルの計算量に対して

$$1 + 1 + (1 + T/B) = 3 + T/B$$

の比で影響を与える。例えば、 $B = 32$  とした場合、 $T = 24$  から  $T = 15$  に変更すると計算量を約 7.5% 削減することができる。したがって、ブロック境界におけるバックワード処理のマージン時点数  $T$  を小さくすることが復号処理における計算量の削減に重要である。

## 【0 0 5 4】

本発明は上記したような従来の技術が有する問題点を解決するためになされたものであり、繰り返し復号を行う復号処理に軟入力軟出力復号を適用する場合に、ブロックの境界におけるバックワード処理のマージン時点数  $T$  を小さく設定しても復号特性の劣化が生じない誤り訂正符号復号方法及び装置を提供することを目的とする。



## 【 0 0 5 5 】

## 【課題を解決するための手段】

上記目的を達成するため本発明の誤り訂正符号復号方法は、所定の復号処理を繰り返すことで受信語を復号する誤り訂正符号復号方法であって、

前記受信語をブロックに分割し、

前記ブロックへの分割方法は、前記所定の復号処理の繰り返しが何回目かであるかに応じて決定し、

前記ブロック単位で、前記受信語及び前記受信語の情報シンボルの信頼度に相当する事前分布値から前記情報シンボルの軟出力値をそれぞれ生成する方法である。

## 【 0 0 5 6 】

ここで、前記ブロックの基準の大きさである時点数を  $B$ 、ブロックの種類数を  $n$ 、現時点の復号処理における繰り返し回数を  $k$  回目としたとき、

前記ブロックへの分割法は、前記ブロックの中で最初のブロックの大きさを、  
 $B(n+1-k)/n$ 、

とし、最終ブロックを除く残りのブロックの大きさを  $B$  としてもよく、

現時点の復号処理における繰り返し回数が奇数回目または偶数回目のいずれか一方で、

最初のブロックの大きさを所定の大きさの  $1/2$  に設定してもよい。

## 【 0 0 5 7 】

一方、本発明の誤り訂正符号復号装置は、所定の復号処理を繰り返すことで受信語を復号する誤り訂正符号復号装置であって、

前記受信語を格納する受信値格納装置と、

前記受信語の情報シンボルの信頼度に相当する事前分布値をそれぞれ保持する事前分布値格納装置と、

前記受信格納装置及び前記事前分布値格納装置から前記ブロック単位でデータを読み出す入力制御装置と、

前記入力制御装置で読み出された前記受信語と前記事前分布値から前記情報シンボルの軟出力値を前記ブロック単位で生成する軟出力生成装置と、

前記入力制御装置に、最初のブロックの大きさを現時点の復号処理における繰り返し数に応じてそれぞれ異なった値に設定させる制御情報生成装置と、  
を有する構成である。

【 0 0 5 8 】

ここで、前記制御情報生成装置は、

前記ブロックの基準の大きさである時点数を  $B$ 、ブロックの種類数を  $n$ 、現時点の復号処理における繰り返し回数を  $k$  回目としたとき、

前記ブロックへの分割法は、前記ブロックの中で最初のブロックの大きさを、  
 $B(n+1-k)/n$ 、

とし、最終ブロックを除く残りのブロックの大きさを  $B$  としてもよく、

前記入力制御装置に、現時点の復号処理における繰り返し数が奇数回目または偶数回目のいずれか一方で、最初のブロックの大きさを所定の大きさの  $1/2$  に設定させてもよい。

【 0 0 5 9 】

上記のような誤り訂正符号復号方法及び装置では、繰り返し処理における各ブロックの種類毎の最初の大きさを、現時点の復号処理における繰り返し数に応じてそれぞれ異なった値に設定することで、ブロックの境界におけるバックワード処理のマージンを小さく設定しても、いずれかのブロックで復号可能であれば、繰り返し復号処理の回数を増加させずに誤りのないデータを高い確率で復号することが可能になる。

【 0 0 6 0 】

【発明の実施の形態】

次に本発明について図面を参照して説明する。

【 0 0 6 1 】

図 1 は本発明の誤り訂正符号復号装置の一構成例を示すブロック図である。

【 0 0 6 2 】

図 1 に示すように、本実施形態の誤り訂正符号復号器は、受信データを格納する受信値格納装置 10 と、情報シンボルの信頼度に相当する事前分布値を保持する事前分布値格納装置 11 と、受信データと事前分布値から各情報シンボルの軟

出力値をブロック単位で生成する軟出力生成装置 13 と、受信値格納装置及び事前分布値格納装置からデータをブロック単位で読み出し、軟出力生成装置に供給する入力制御装置 12 と、繰り返し復号処理の回数の応じて入力制御装置にブロックの境界を変更させる制御情報生成装置 14 とを有する構成である。

#### 【0063】

制御情報生成装置 14 は、ブロックを複数種類設け、繰り返し処理における各ブロックの種類毎の最初の大きさを、繰り返し復号処理の回数の応じてそれぞれ異なった値に設定する。例えば、ブロックを 2 種類設けた場合は、繰り返し復号処理の回数が奇数回目、あるいは偶数回目のいずれか一方の種類の最初のブロックの大きさを基準の大きさ  $B$  と異なった大きさ（例えば、 $B/2$ ）に設定する。

#### 【0064】

次に、本実施形態の誤り訂正符号復号器の動作について図 2 及び図 3 を用いて具体的に説明する。図 2 は図 1 に示した制御情報生成装置の処理手順の一例を示すフローチャートであり、図 3 は図 2 に示したフローチャートにしたがって受信データが復号処理される様子を示す模式図である。なお、以下ではブロックを 2 種類設けた場合を例にして説明する。

#### 【0065】

図 2 において、制御情報生成装置 14 は、まず軟出力生成を終了していない残りのデータの時点数（残データ時点数） $R$ を確認する（ステップ A1）。残データ時点数  $R$  が  $B$  よりも小さい場合はブロックの大きさを  $R$  に設定する（ステップ A6）。また、残データ時点数  $R$  が  $B$  以上の場合は軟出力生成対象のデータが最初のブロックであるか否かを判定する（ステップ A2）。ステップ A2 の処理の結果、軟出力生成対象のデータが最初のブロックでない場合はブロックの大きさを  $B$  に設定する（ステップ A5）。また、最初のブロックの場合は現在の繰り返し復号処理の回数が偶数回目であるか奇数回目であるかを判定する（ステップ A3）。繰り返し数が偶数回目である場合は最初のブロックの大きさを  $B/2$  に設定する（ステップ A4）。また、繰り返し数が奇数回目である場合は最初のブロックの大きさを  $B$  に設定する（ステップ A5）。

#### 【0066】

図2に示した制御情報生成装置14の動作によれば、繰り返し復号処理の回数が奇数回目の場合は図3の(1)に示すようにブロックの境界が設定され、繰り返し復号処理の回数が偶数回目の場合は図3の(2)に示すようにブロックの境界が設定され、そのブロック単位で軟出力生成処理が行われる。つまり、繰り返し復号処理の奇数回目と偶数回目とではブロックの境界が $B/2$ 時点数だけずれることになる。なお、奇数回目と偶数回目におけるブロック境界の設定は逆であってもよい。

## 【0067】

上述したように、従来の復号方法では、マージン時点数 $T$ を小さく設定した場合に復号性能の劣化が生じ、 $B$ 、 $T$ の値が同一でもエラーパターンによってはブロック境界の設定の仕方で復号が可能な場合とそうでない場合が発生する。なお、二つのブロック境界を用いて復号を行い、正しく復号できたと判断した方を復号結果とする復号方法も考えられるが、これでは逆に計算量が大きくなってしまふ。

## 【0068】

本実施形態のようにブロック境界を繰り返し処理毎に変更することでマージン時点数 $T$ を小さく設定しても計算量を増加させずに復号性能の向上が期待できる。

## 【0069】

以上、繰り返し復号処理を行う復号過程において2種類のブロックを交互に使用する例で説明したが、この方法は2種類のブロックに限らず3種類以上のブロックを使用する場合にも容易に拡張できる。

## 【0070】

図4を用いて $n$ 種類のブロック境界を用いた場合の誤り訂正符号復号装置の動作について説明する。図4は図1に示した制御情報生成装置の処理手順の他の例を示すフローチャートである。

## 【0071】

図4に示すように、制御情報生成装置14は、まず軟出力生成を終了していない残りのデータの時点数(残データ時点数) $R$ を確認する(ステップB1)。残

データ時点数  $R$  が  $B$  より小さい場合はブロックの大きさを  $R$  に設定する(ステップ  $B5$ )。また、残データ時点数  $R$  が  $B$  以上の場合は軟出力生成対象のデータが最初のブロックであるか否かを判定する(ステップ  $B2$ )。軟出力生成対象のデータが最初のブロックでない場合はブロックの大きさを  $B$  に設定する(ステップ  $B4$ )。また、軟出力生成対象のデータが最初のブロックの場合は現在の繰り返し回数を確認し、繰り返し回数が  $k$  回目の場合はブロックの大きさを

【0072】

【数7】

$$B(n+1-k)/n \quad \dots(3)$$

【0073】

に設定する(ステップ  $B3$ )。

【0074】

図5は図1に示した制御情報生成装置の処理結果を示す図であり、ブロックの大きさの基準を  $B$  としたときの  $n$  個のブロックの境界が設定される様子を示す模式図である。

【0075】

図4に示した手順で繰り返し処理の回数が  $k$  回目におけるブロック境界の設定は図5に示すブロック境界  $(k \bmod n)$  に対応する。ここで、 $k \bmod n$  は  $n$  を法とする  $k$  の値を示し、 $k$  は1から  $n$  までの値でとるものとする。

【0076】

次に、本実施形態の誤り訂正符号復号器を図7に示したターボ復号器に適用した場合の動作について図6を用いて説明する。図6は図2に示した処理をターボ復号器に適用した場合の処理手順を示すフローチャートである。なお、図7に示した第1の復号器、第2の復号器は図1に示した誤り訂正符号復号器で構成されているものとする。また、以下では、図2、3に示した2種類のブロックを設ける場合を例にして説明するが、図4、図5に示した3種類以上のブロックを用いる場合も同様に実現できる。なお、受信データは第1の復号器103の受信値格納装置に格納されているとする。また、繰り返し復号処理の回数に対応してプロ

ックの境界を変更する処理は第 1 の復号器の制御情報生成装置で行うものとする。

#### 【 0 0 7 7 】

最初に、第 1 の復号器の制御情報生成装置は、繰り返し復号処理の回数をカウントするための変数である  $k$  を 1 に設定する(ステップ C 1)。続いて、 $k$  の値が奇数であるか偶数であるかを判定し(ステップ C 2)、偶数と判定した場合は図 3 (2) に示すようにブロックの境界を設定し(ステップ C 3)、奇数と判定した場合は図 3 (1) に示すようにブロックの境界を設定するように(ステップ C 4)、入力制御装置に指示する。

#### 【 0 0 7 8 】

入力制御装置は制御情報生成装置の指示にしたがって受信値格納装置及び事前分布格納装置からブロック単位でデータを読み出し、軟出力復号装置に供給する。

#### 【 0 0 7 9 】

以上の処理は、先ず最初にターボ符号の構成要素である第 1 の畳込み符号化器に対応する第 1 の復号器で行う(ステップ C 5)。このとき第 1 の復号器ではステップ C 2 で指定されたブロック単位で軟出力を生成する。

#### 【 0 0 8 0 】

$k = 1$  の場合、事前分布格納装置には適切な初期値(通常は 0, 1 を等確率であるとした値)が設定される。第 1 の復号器の復号結果である外部情報は第 2 の復号器の事前分布格納装置に保持される。この外部情報も合わせて利用することで第 2 の復号器で復号処理を実行する(ステップ C 6)。このとき第 2 の復号器ではステップ C 2 で設定されたブロック単位で軟出力を生成する。ステップ C 6 の処理により第 2 の復号器から出力された外部情報は第 1 の復号器の事前分布格納装置に保持される。

#### 【 0 0 8 1 】

以上の処理が終了したら、第 1 の復号器の制御情報生成装置は、繰り返し復号処理を継続するか否かを判定する(ステップ C 7)。これは、予め決められた繰り返し回数に達したか否かを判定する方法、あるいは外部に付加した誤り検出符

号をチェックして判定する方法など様々な方法が考えられる。繰り返し処理を継続する場合は $k$ の値を1インクリメントし（ステップC8）、ステップC2の処理に戻ってステップC7までの処理を繰り返す。

#### 【0082】

したがって、繰り返し処理における各ブロック種類毎の最初の大きさを、繰り返し復号処理の回数の応じてそれぞれ異なった値に設定することで、ブロック境界におけるバックワード処理のマージン時点数 $T$ を小さく設定しても、いずれかのブロックで復号可能であれば、繰り返し復号処理の回数を増加させずに誤りのないデータを高い確率で復号することが可能になる。よって、復号性能の劣化を生じることなく、バックワード処理の計算量を削減できる。

#### 【0083】

例えば、メモリ数3、符号化率 $1/3$ のターボ符号において、ブロックの大きさ $B=32$ とし、符号トレリス全体を持って復号する場合と比較した場合、復号フレーム誤り率 $0.0001$ において、従来のブロック境界を固定する方法では $T=18$ で $0.1\text{ dB}$ 以内の符号化利得の差となるが、本実施形態のブロック境界を3種類用いる方法では同一の符号化利得を $T=12$ で達成できる。これは5%程度の計算量の削減となる。

#### 【0084】

##### 【発明の効果】

本発明は以上説明したように構成されているので、以下に記載する効果を奏する。

#### 【0085】

ブロックを複数種類設け、繰り返し処理における各ブロックの種類毎の最初の大きさを、現時点の復号処理における繰り返し数に応じてそれぞれ異なった値に設定することで、ブロックの境界におけるバックワード処理のマージンを小さく設定しても、いずれかのブロックで復号可能であれば、繰り返し復号処理の回数を増加させずに誤りのないデータを高い確率で復号することが可能になる。

#### 【0086】

したがって、復号性能の劣化を生じることなく、バックワード処理の計算量を

削減できる。

【図面の簡単な説明】

【図 1】

本発明の誤り訂正符号復号装置の一構成例を示すブロック図である。

【図 2】

図 1 に示した制御情報生成装置の処理手順の一例を示すフローチャートである。

【図 3】

図 2 に示したフローチャートにしたがって受信データが復号処理される様子を示す模式図である。

【図 4】

図 1 に示した制御情報生成装置の処理手順の他の例を示すフローチャートである。

【図 5】

図 1 に示した制御情報生成装置の処理結果を示す図であり、ブロックの大きさの基準を B としたときの n 個のブロックの境界が設定される様子を示す模式図である。

【図 6】

図 2 に示した処理をターボ復号器に適用した場合の処理手順を示すフローチャートである。

【図 7】

ターボ符号器、復号器の代表的な構成を示す図であり、同図 (a) はターボ符号器のブロック図、同図 (b) はターボ復号器のブロック図である。

【図 8】

従来の誤り訂正符号復号装置の構成を示すブロック図である。

【図 9】

ブロック毎に軟出力を生成する従来の誤り訂正符号復号装置の処理手順を示すフローチャートである。

【図 10】



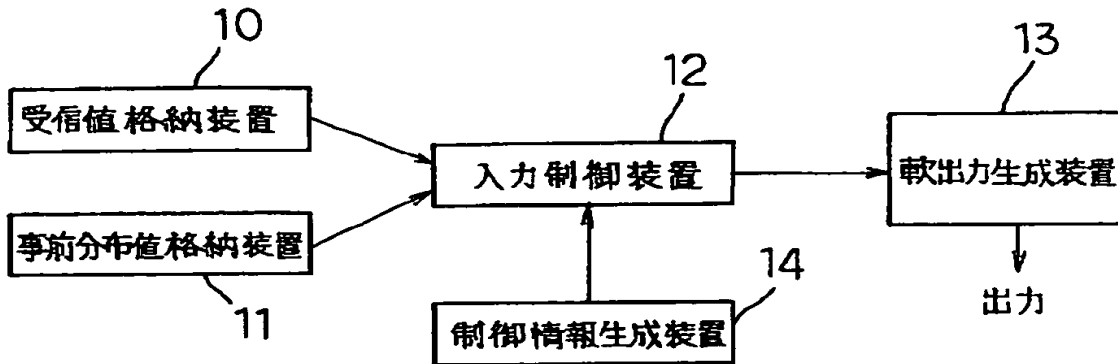
図 9 に示したフローチャートにしたがって受信データが復号処理される様子を  
示す模式図である。

【符号の説明】

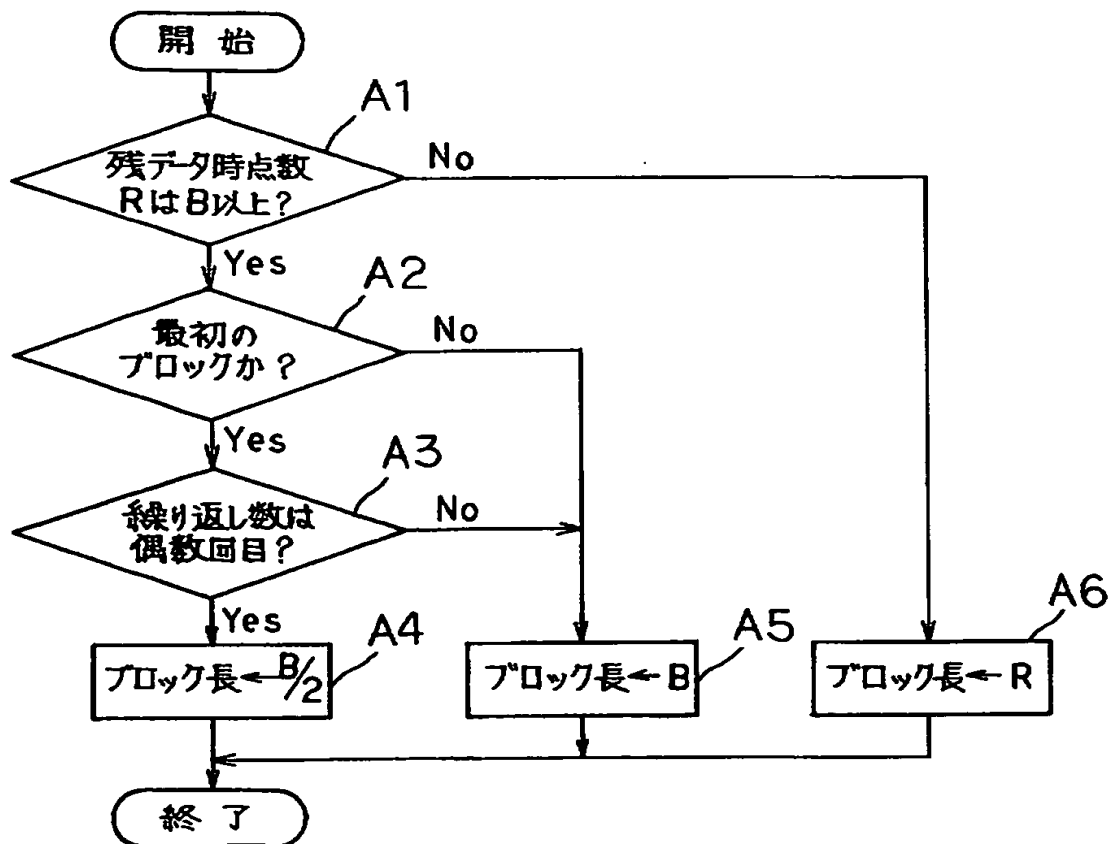
- 1 0 受信値格納装置
- 1 1 事前分布値格納装置
- 1 2 入力制御装置
- 1 3 軟出力生成装置
- 1 4 制御情報生成装置

【書類名】 図面

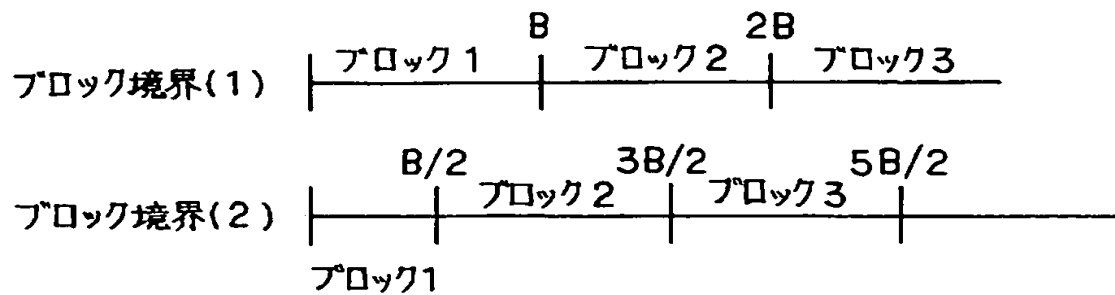
【図1】



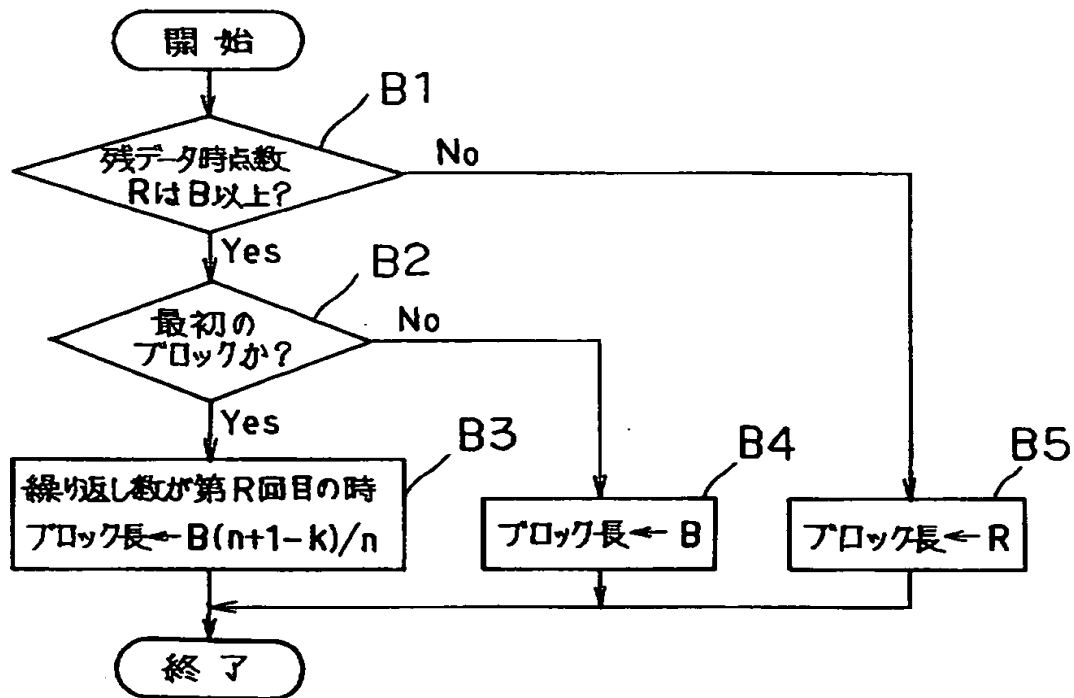
【図2】



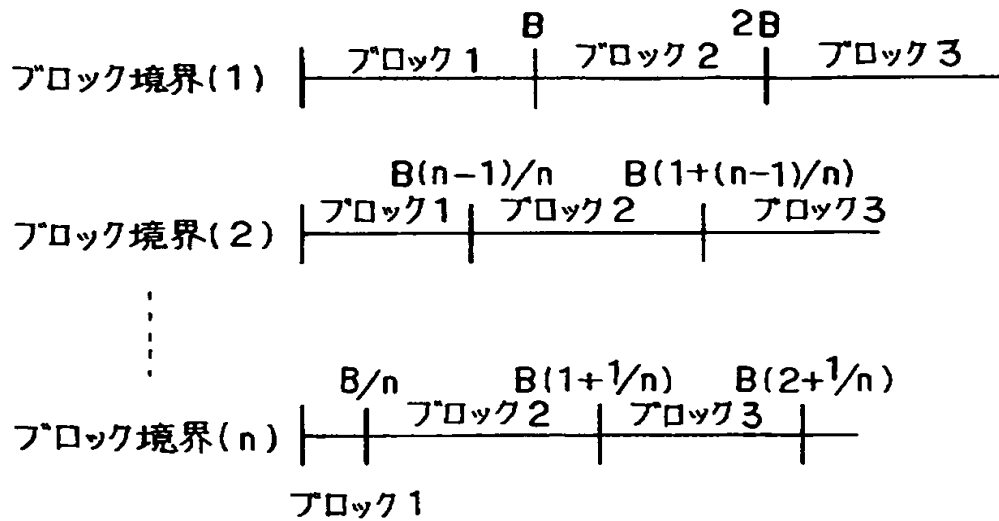
【図3】



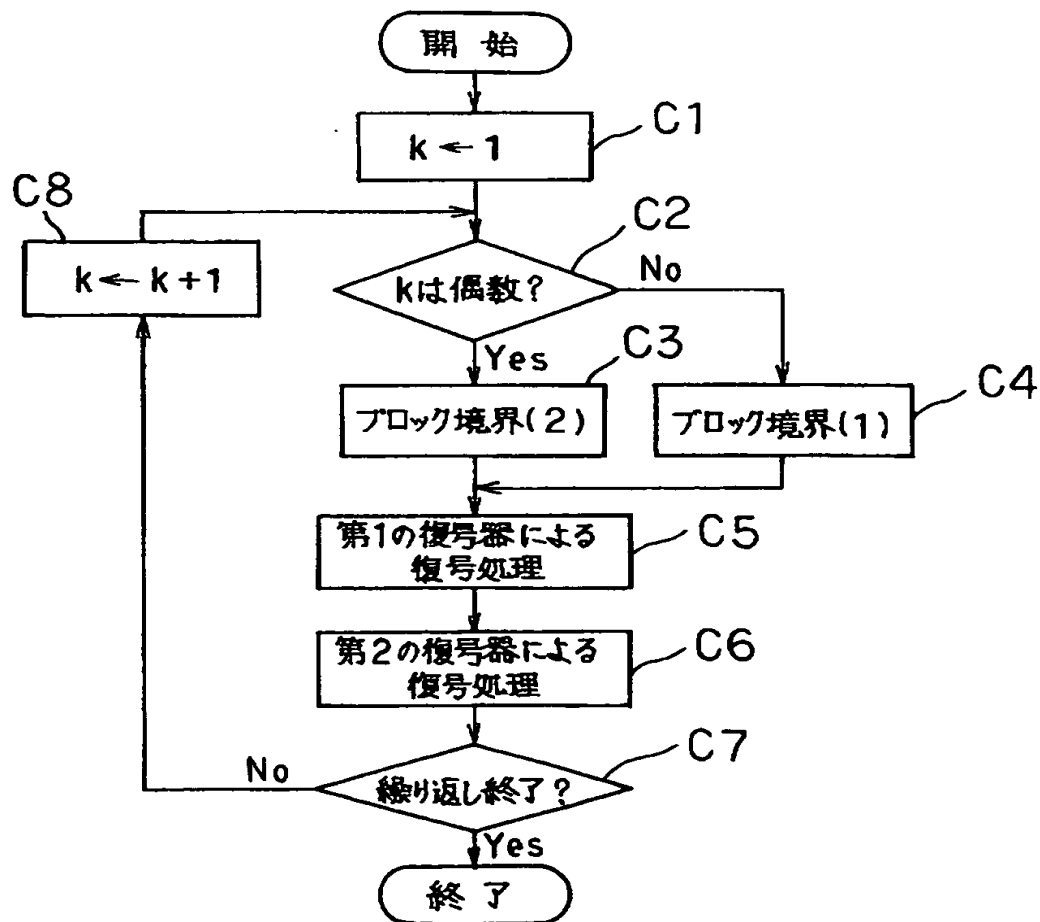
【図4】



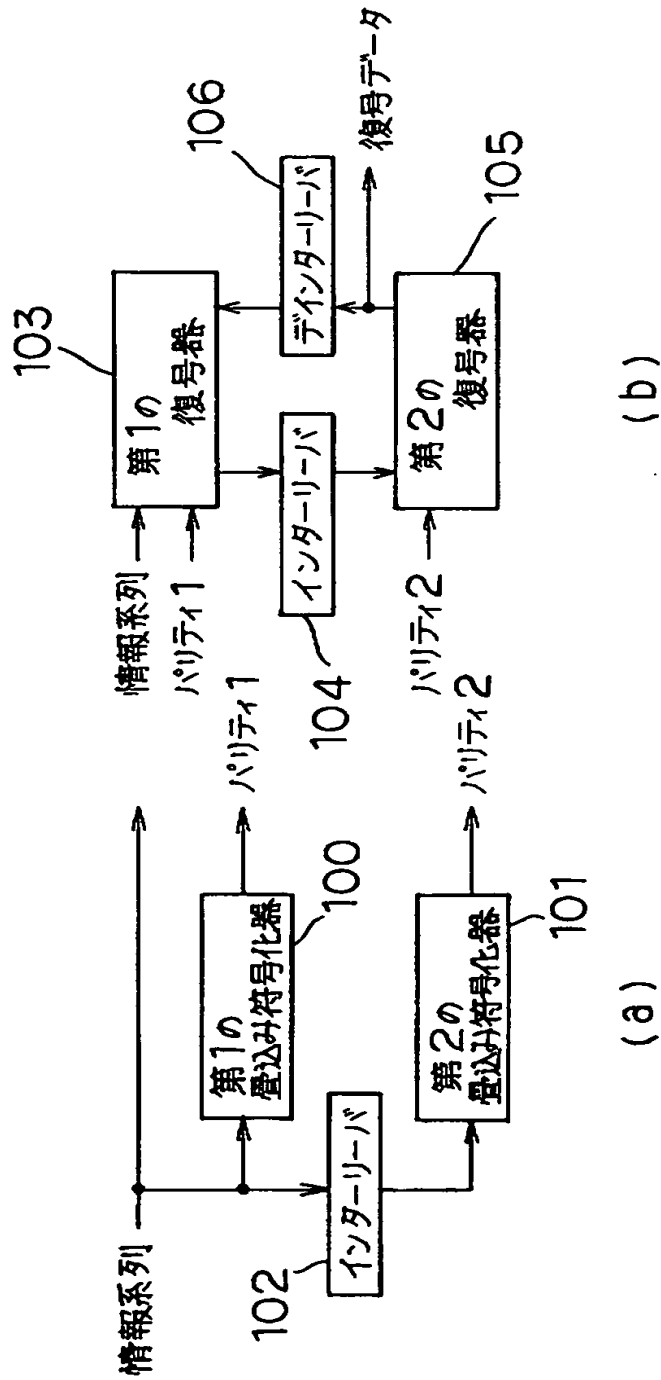
【図5】



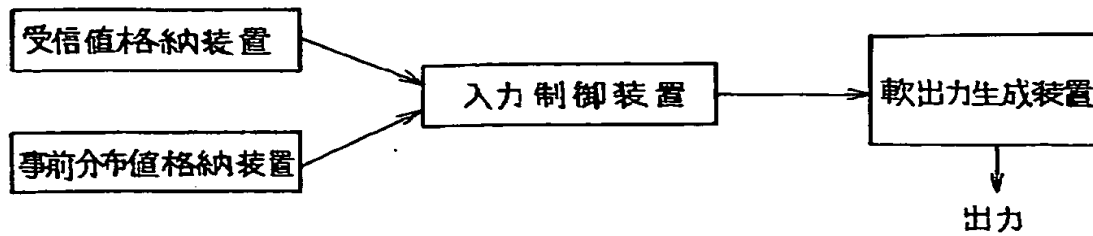
【図6】



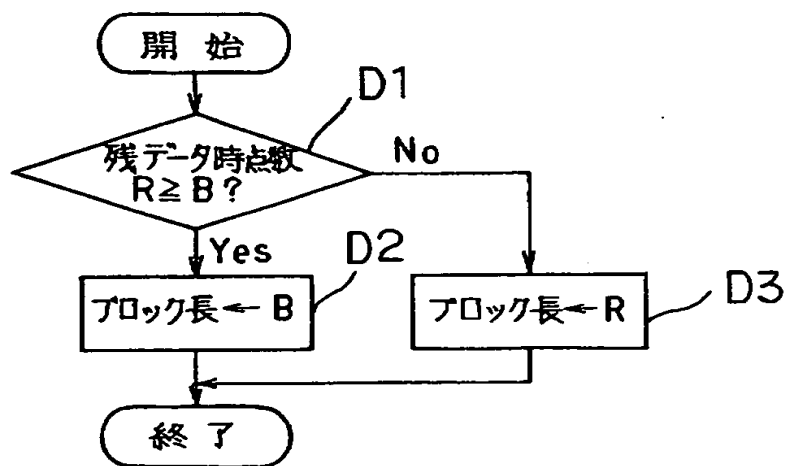
【図7】



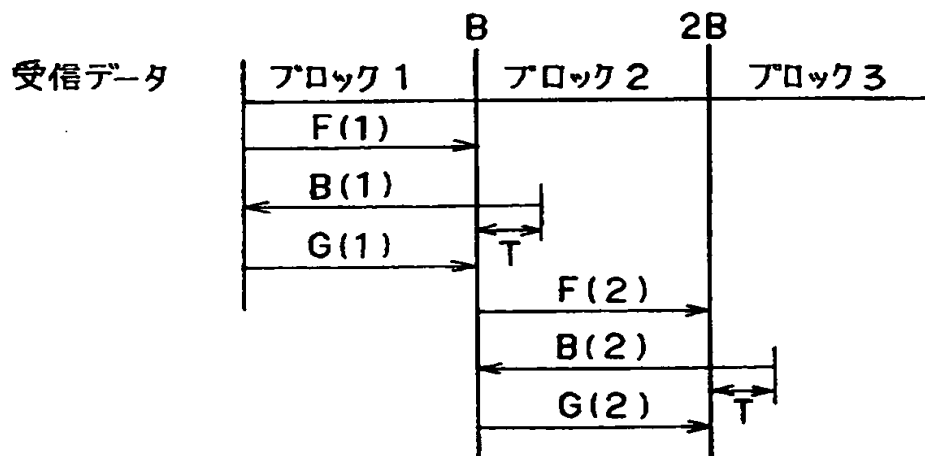
【図 8】



【図 9】



【図 10】



【書類名】 要約書

【要約】

【課題】 繰り返し復号を行う復号処理に軟入力軟出力復号を適用する場合に、ブロックの境界におけるバックワード処理のマージンを小さく設定しても復号特性の劣化が生じない誤り訂正符号復号方法及び装置を提供する。

【解決手段】 所定の復号処理を繰り返すことで受信語を復号する誤り訂正符号復号方法であって、受信語をブロックに分割し、ブロックへの分割方法は、所定の復号処理の繰り返しが何回目かであるかに応じて決定し、ブロック単位で、受信語及び受信語の情報シンボルの信頼度に相当する事前分布値から情報シンボルの軟出力値をそれぞれ生成する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社